DevOps Technical Assessment Duration: 2-3 hours (+ 1-2 hours for bonus)

Task Overview: Create a simple containerized web application with frontend and backend services.

**Core Requirements:**

1. Application Components:
    - Create a simple frontend (any framework of your choice - React, Vue, Angular)
        - Single page showing a "Hello World" message
        - Should make an API call to backend
    - Create a basic backend (any language/framework of your choice)
        - Single endpoint returning a simple JSON response

2. Docker Requirements:
    - Create Dockerfiles for both frontend and backend
    - Implement multi-stage builds where appropriate
    - Create a docker-compose file to run both services
    - Services should be able to communicate with each other
    - Include appropriate environment variables

3. Build Process:
    - Create a build script to automate the build process
    - Include basic error handling in the build process
    - Add appropriate logging/output during build

4. Documentation Requirements:
    - README file with:
        - Setup instructions
        - Requirements list
        - Architecture decisions explanation
        - Local development guide
        - Troubleshooting section

**Monitoring Requirement:**

1. Application Instrumentation:
    - Add Prometheus metrics to both services:
        - Frontend: Basic metrics like page loads, API call latency

        - Backend: Request counts, response times, error rates
        - Custom metrics of your choice that you think are valuable

2. Monitoring Stack:
    - Add Prometheus server to collect metrics
    - Configure Grafana for visualization
    - Create a basic dashboard showing:
        - Service health/uptime
        - Performance metrics
        - Error rates
        - Custom metrics visualization

3. Infrastructure Requirements:
    - Add Prometheus and Grafana to your docker-compose setup
    - Configure proper service discovery
    - Ensure persistent storage for Grafana dashboards
    - Set up proper network connectivity between all components

4. Documentation Additions:
    - Document all custom metrics
    - Include dashboard setup instructions
    - Provide sample queries for important metrics
    - Add monitoring architecture explanation

Submission:

- Provide a Git repository with your solution
- Include all source code, configuration files, and documentation
- Ensure all instructions are clear and complete

This assessment will evaluate:

- Container knowledge
- Automation skills
- Documentation abilities
- Understanding of service communication
- Security awareness
- Problem-solving approach

- Observability principles (bonus)
- Monitoring system configuration (bonus)
- Metric design knowledge (bonus)
- Integration capabilities (bonus)